

LETTER TO THE EDITOR

Dear CROSSTALK Editor,

It will become increasingly clear to policy-makers over the next several years that the open source paradigm must be embraced by the Department of Defense (DoD). Despite the compelling arguments for open source and broad commercial industry acceptance, there will be significant government structural and defense industry interest impeding the inevitable. These impediments will likely only fall when one or more major defense acquisition programs (MDAPs) fail due to software cost overruns and schedule delays, affecting the bottom line of major defense contractors. However, determined policy efforts could accelerate the adoption of open source, saving the DoD billions of dollars, reducing delays, and facilitating joint operations through interoperability.

The size of the effective source lines of code (SLOC) for MDAPs has reached well over 10 million SLOC and is growing rapidly. The development cost of this software, which the Office of the Secretary of Defense (OSD) Cost Analysis Improvement Group (CAIG) puts at more than \$300/SLOC, is only decreasing slowly as increasing complexity competes with process and tool improvements. As a result, software development is quickly approaching 50 percent of the cost of MDAP development, and the eight years or more of development schedule has placed software permanently on the critical path to deployment. Software has become the cost- and schedule-limiting factor of our future defense capabilities.

Many of the requirements driving software have and are being addressed by many programs, yet data suggests only a (roughly) 30 percent savings is achieved by reusing and modifying existing code. Code reused from other programs must often be substantially modified to adapt to the architecture and specific requirements of another program. Architectural differences between programs impede coarse-grained reuse while differences in the details of requirements require modification of code at the fine grain. Reuse is further hampered by the concurrent evolution of the programs being reused. Programs committed to reuse as a cost saving measure can establish program-to-program memorandums of agreement, but it is easy to see how this would become an n-factorial interface coordination problem when looking across multiple programs, with significant duplication of effort.

A current alternative approach to open source is for software reuse to be mandated by policy and the establishment of joint development efforts such as the Mission Planning System Framework and Common Capabilities being developed by the Air Force and Navy. Unfortunately, *not-invented-here* concerns have prevented widespread assumption of this important command-and-control software initiative. As far as I know, no MDAPs have actively participated in the requirements definition of the Common Capabilities or adapted their command and control architecture to leverage or influence this development.

The open source development paradigm offers the

promise and presents the challenges of coordinated definition of requirements, architecture and design approach, coding, and testing. The promise is enhanced reuse and interoperability, shared development costs, and shortened development schedules. The challenges, among others, are security, organizational and contractual performance responsibility, and proprietary and licensing considerations. These challenges should be studied and can be overcome. Many of them have been and are being addressed in a variety of ways by the commercial open source community, which now includes the largest commercial software vendors, including IBM, Novell, Sun Microsystems, and even Microsoft.

The development and advocacy of modern and flexible open architectures and standards by the Defense Information Systems Agency is a necessary precondition for guiding the development of potentially hundreds of open source projects within the DoD so that the software for multiple MDAPs can be more easily assembled from open source code. Once security and other key issues have been addressed, I recommend the establishment of (including ground rules for) a repository for DoD open source software development, modeled in some ways along the lines of the popular <www.sourceforge.net> open source development Web site. Perhaps the URL might be <www.sourceforge.mil>. The force of arguments for open source are so compelling that early adopters in the DoD will find many reasons to utilize such a repository once it becomes available. Among other development artifacts, requirements databases, unified modeling language models, source code, and bug resolution tracking should be high priorities for such a repository.

In the meantime, I encourage the DoD software acquisition community to become more aware of the open source development. Junior officers and developers might want to select one of the more than 50,000 projects at <www.sourceforge.net> to participate in. Programming skills are not necessarily required. One can contribute in many ways, including software testing and documentation. Software acquisition managers could consider how to leverage existing open source software in the programs they manage. Policy-makers could engage industry leaders that have recognized the marketplace advantages of open source to address unique DoD concerns. I hope that open source advocates will spread across the software acquisition community and important DoD software institutions that include Carnegie Mellon University's Software Engineering Institute, the U.S. Air Force's Software Technology Support Center, and the MITRE Corporation.

Finally, I hope that the defense industry will recognize that the open source paradigm can be tailored and leveraged into increased competitiveness, productivity, and profits, while delivering more capability to our military faster and cheaper.

Thomas M. Schaefer
Senior Defense Cost Analyst and Software Developer